

Displays gebruiken in Raspberry Pi-projecten

Een hoofdstuk als voorbeeld: Organische LED-displays (OLED)

Dogan Ibrahim (Verenigd Koninkrijk)

In deze aflevering van *Elektor Books* presenteren we een hoofdstuk uit Dogan Ibrahim's *Elektor-boek Using Displays in Raspberry Pi Projects*. OLED-displays zijn al sinds hun introductie favoriet bij makers en ontwerpers van embedded systemen. In dit artikel onderzoeken we manieren om OLED-displays te gebruiken in projecten met zo weinig mogelijk (programmeer)inspanning. Het spreekt bijna vanzelf dat we dan meteen aan de Raspberry Pi denken. We gaan meteen aan de slag!



Figuur 1: Enkele OLED-displays.

Opmerking van de redactie: Dit artikel is een hoofdstuk uit het boek Using Displays in Raspberry Pi Projects, dat we enigszins hebben bewerkt om het in het formaat van Elektor Magazine te passen. Omdat dat een gedeelte is uit een grotere publicatie, kunnen er in dit artikel verwijzingen voorkomen naar andere hoofdstukken in het boek. De auteur en de redactie hebben hun best gedaan om dit te vermijden, en zijn altijd bereid om op vragen in te gaan. U vindt hun contactgegevens in het tekstkader Vragen of opmerkingen?

Een organisch LED-display bevat een elektro-luminescente laag van organisch materiaal die licht uitstraalt wanneer er een elektrische stroom door loopt. De organische laag bevindt zich tussen twee elektroden, waarvan er tenminste één transparant is. OLED-displays worden gebruikt in veel commerciële toepassingen, zoals TV-schermen, computerschermen, smartphones, spelconsoles en andere digitale displays. OLED's kunnen worden aangestuurd met een passieve (PMOLED) of een actieve matrix (AMOLED). Bij PMOLED-displays worden de rijen en kolommen sequentieel aangestuurd. Bij AMOLED-displays wordt een

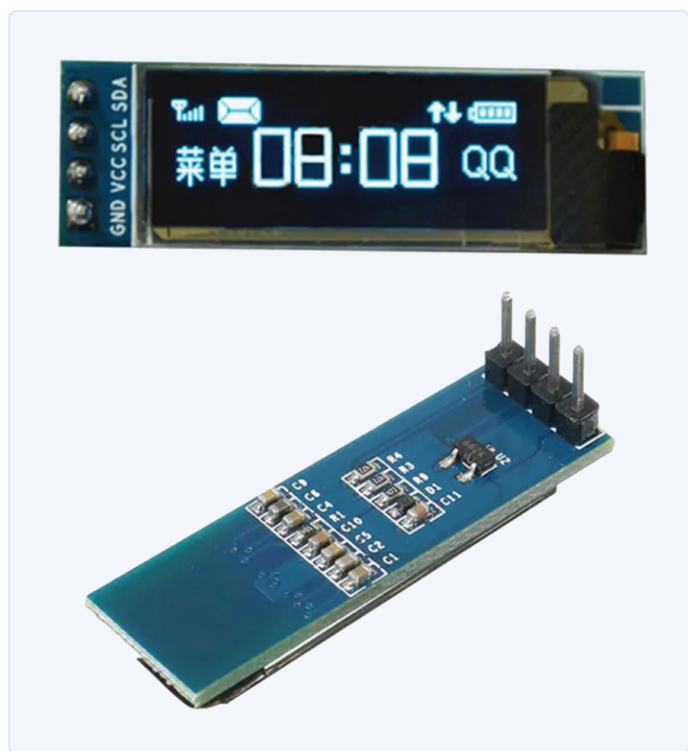
dunne filmtransistor gebruikt om elk individueel pixel aan of uit te zetten.

OLED-displays hebben over het algemeen de volgende voordelen in vergelijking met LCD's:

- OLED's gebruiken erg weinig stroom, wat ze aantrekkelijk maakt voor mobiele toepassingen.
- OLED's bieden een betere beeldkwaliteit, met beter contrast, meer helderheid, een grotere zichthoek en een snellere refresh-tijd.
- OLED-displays zijn robuuster, wat ze geschikt maakt voor een groter temperatuurbereik.
- OLED-displays kunnen ook buigzaam worden geproduceerd. Ze zijn dan bijvoorbeeld opvouwbaar, om de pols te dragen, enzovoort.

Maar OLED-displays zijn niet perfect, ze hebben ook nadelen:

- Ze zijn aanzienlijk duurder dan LCD's.
- Ze hebben een beperkte levensduur, hoewel die de laatste tijd steeds beter wordt.



Figuur 2: 128 x 32-pixel OLED. Bron: <https://uk.banggood.com>

- OLED-displays zijn in direct zonlicht soms slecht af te lezen omdat ze zelf licht uitstralen

In **figuur 1** ziet u enkele voorbeelden van OLED-displays.

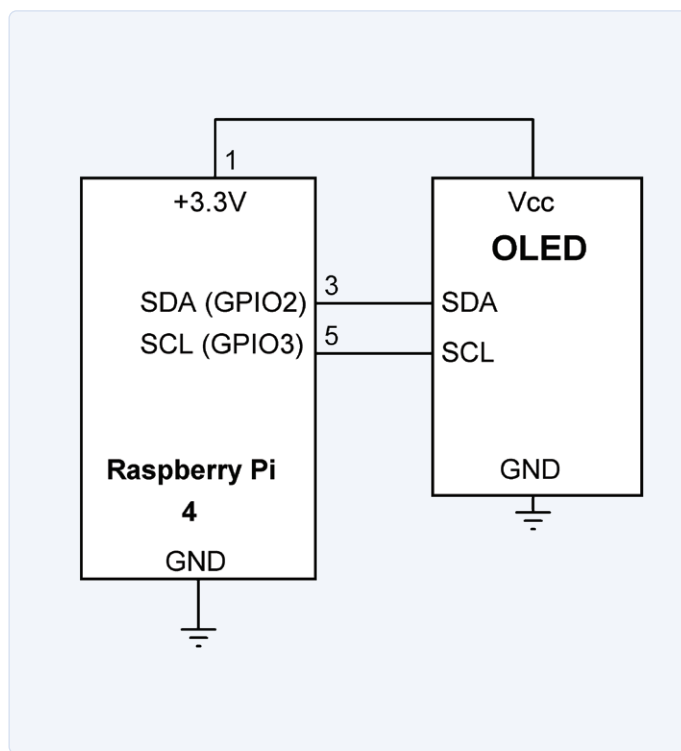
Toepassen van OLED-displays

Er zijn OLED-displays in allerlei afmetingen met verschillende pixelmatrices. Natuurlijk kunnen we niet alle typen OLED-displays bespreken in dit artikel. We zullen ons hier beperken tot het zeer populaire OLED-display met 128 × 32 pixels, dat wordt bestuurd met een SSD1306-chip. Het is de moeite waard om de eigenschappen van dit display te bestuderen, voordat u het inzet in uw projecten. In **figuur 2** ziet u het OLED-display dat we gaan gebruiken.

De specificaties zijn als volgt:

- Resolutie: 128 × 32 pixels
- Pixelkleur: wit (blauw is ook verkrijgbaar)
- Grootte: 0,91 inch
- Interface: I²C
- Driver: SSD1306
- 4 pennen: GND, VCC, SCL (I²C), SDA (I²C)
- Zichthoek: Meer dan 160 graden
- Werkspanning: +3,3 V / +5 V

Het OLED-display met SSD1306-driver wordt aan de host gekoppeld via de I²C interface (er worden dus maar 2 pennen gebruikt voor de interface, daarnaast zijn er één pen voor de voeding en één voor de massa). Het I²C-adres van de driver is standaard 0x3C. In **figuur 3** ziet u de verbinding tussen de Raspberry Pi en het OLED-display. De chip werkt normaal gesproken op +3,3 V, maar er zit een spanningsregelaar op de chip, waardoor hij ook met +5 V gevoed mag worden.



Figuur 3: Het schema.

Raspberry Pi-poort	Raspberry Pi-pen	OLED-pen
SDA	(GPIO2)	2
SCL	(GPIO3)	3
GND	39	GND
+3.3V	1	V _{cc}

Installeren van de OLED-library

De I²C-interface van de Raspberry Pi moet worden ingeschakeld met behulp van het `sudorasp-config` configuratietool. Daarna moet u de OLED-library installeren, we gebruiken hier de library `Adafruit_Python_SSD1306`.

De stappen zijn als volgt (sommige van deze bibliotheken zijn misschien al geïnstalleerd op uw Raspberry Pi):

```
pi@raspberrypi:~$ sudo apt install -y python3-dev
python3-pil python3-pip python3-setuptools python3-rpi.gpio
pi@raspberrypi:~$ sudo pip3 install
adafruit-circuitpython-ssd1306
```

Voordat we een apparaat via de I²C-interface kunnen aansturen, moeten we eerst het I²C-adres detecteren. Geef het volgende commando om het I²C-adres van het display te detecteren:

```
pi@raspberrypi:~$ i2cdetect -y 1
```

Het resultaat zou er uit moeten zien als in **figuur 4**.

```

pi@raspberrypi:~ $ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  3c  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~ $ █

```

Figuur 4: Detecteren van het OLED-display.



Figuur 5: Coördinaten van het scherm.



Listing 1: OLEDorners.py.

```

#-----
#
#      SHOW THE PIXELS AT CORNERS OF THE DISPLAY
#      =====
#
# This program shows the pixels at the four corners of the display
#
# Author: Dogan Ibrahim
# File  : OLEDcorners.py
# Date  : November 2020
#-----
#
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)

display.fill(0)          # Clear pixels
display.show()           # Display data

display.pixel(0,0,1)      # Pixel at (0,0)
display.pixel(127,0,1)    # Pixel at (127,0)
display.pixel(0,31,1)     # Pixel at (0,31)
display.pixel(127, 31, 1)  # Pixel at (127,31)
display.show()            # Display the data

```

Project 1: Pixels weergeven in de vier hoeken van het scherm

De coördinaten van het OLED-display zijn te zien in **figuur 5**. De linker bovenhoek is (0, 0). De X-coördinaat telt van links naar rechts van 0 tot 127, de Y-coördinaat telt van boven naar beneden van 0 tot 31.

Programma-listing: Het programma *OLEDcorners.py* is weergegeven in **listing 1**. Dit programma zit in het software-archiefbestand voor het boek en is te vinden op [1] onder *Downloads*. Aan het begin van het programma wordt de Adafruit OLED-library geïmporteerd. Dan worden de pixels gewist met de functie-aanroep `display.fill(0)`. Het is aan te raden om altijd de pixels te wissen bij het begin van een programma. Daarna worden de pixels op de vier hoeken van het display aangezet. Met de functie `display.show()` wordt de data weergegeven op het display.

In **figuur 6** ziet u het display met de vier hoekpixels ingeschakeld.

Project 2: Tekst weergeven

Bij dit project geven we de tekst 'ELEKTOR' weer vanaf coördinaat (15, 5) van het display.

Programma-listing: In **listing 2** ziet u het programma *OLEDText.py*. Het maakt gebruik van de PIL-library-functies [2]. Aan het begin van het programma worden de image-en font-modules van de PIL-bibliotheek geïmporteerd. Dan wordt de default font-library geladen. Daarna wist het programma de pixels. In `width` en `height` worden de breedte (128) en hoogte (32) van het display vastgelegd.

Om een beeld samen te stellen, moet eerst een leeg beeld worden gemaakt met de afmetingen van het scherm. Dat gebeurt met het statement:

```
image = Image.new('1', (width, height))
```

Met dit image-object wordt een draw-object aangemaakt waarmee vormen en tekst worden getekend:

```
draw = ImageDraw.Draw(image)
```

In dit programma willen we tekst weergeven op de coördinaten (15, 5). Dat doen we met de functie `draw.text`:



Listing 2: OLEDText.py.

```

#-----
#      DISPLAY TEXT
#      =====
#
# This program displays the text E L E K T O R at (15,5)
#
# Author: Dogan Ibrahim
# File  : OLEDText.py
# Date  : November 2020
#-----
from PIL import Image, ImageDraw, ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)
font = ImageFont.load_default() # Default font

display.fill(0)          # Clear display
display.show()

width = display.width      # Width
height = display.height   # Height
image = Image.new('1', (width, height))
draw = ImageDraw.Draw(image)
draw.text((15, 5), "E L E K T O R", font = font, fill = 255)
display.image(image)
display.show()

```

```
draw.text((15, 5), "E L E K T O R", font = font, fill
= 255)
```

`fill = 255` tekent het beeld in het wit `0` tekent het in het zwart. Daarna moeten we het beeld weergeven met de functies `display.image(image)` en `display.show()`. **Figuur 7** toont hoe dat er uitziet.

Project 3: Vormen weergeven

We kunnen verschillende vormen op het display tekenen. Zie voor een compleet overzicht van de functies de documentatie van de PIL-library: er zijn veel meer functies dan we hier in het kort beschrijven.

Rectangle (rechthoek):

(x,y) is de linker bovenhoek van de rechthoek. `Outline` is de kleur van de omtrek (255 voor wit, 0 voor zwart), `fill` is de kleur van de binnenkant van de vorm (255 voor wit, 0 voor zwart).

```
draw.rectangle((x, y, width, height), outline = nn,
fill = mm)
```

Ellipse (ellips):

```
draw.ellipse((x, y, width, height), outline = nn,
fill = mm)
```

Line (lijn):

(x_1,y_1) is de eerste coördinaat, (x_2,y_2) is de tweede coördinaat.

```
draw.line((x1, y1, x2, y2), fill = mm)
```

Arc (boog):

```
draw.arc((x1,x2,y1,y2), s1, s2, fill = mm)
```

Tekent een boog tussen de starthoek (s_1) en eindhoek (s_2), in de rechthoek (x_1,x_2,y_1,y_2) .

Chord (kooorde):

```
draw.chord((x1,x2,y1,y2), s1, s2, fill = mm)
```

Teken een lijn tussen de starthoek (s_1) en eindhoek (s_2), in de rechthoek (x_1,x_2,y_1,y_2) .

PieSlice (taartpunt):

```
draw.pieslice((x1,y1,x2,y2), s1, s2, fill = mm)
```

Hetzelfde als Arc, maar tekent ook rechte lijnen tussen de eindpunten en het centrum van de gegeven rechthoek.

Point (punt):

```
draw.point((x1,x2), fill = mm)
```

Tekent pixels op de gegeven coördinaten. Het is ook mogelijk om meer dan één pixel te tekenen, zoals bij het onderstaande commando waar 3 individuele pixels worden getekend op de gegeven coördinaten:

```
draw.point([(x1,y1), (x2,y2), (x3,y3)], fill = mm)
```

Polygon (polygoon):

```
draw.polygon([(x1,y1), (x2,y2), (x3,y3), .....], outline
= nn, fill = mm)
```

Tekent een polygoon, waarvan de omtrek bestaat uit rechte lijnen tussen de gegeven coördinaten, plus een rechte lijn tussen de laatste en de eerste coördinaten. De lijst van coördinaten kan elk soort object met 2-tuples $[(x, y), \dots]$ of numerieke waarden $[x, y, \dots]$ zijn. Er moeten tenminste drie punten gegeven zijn.



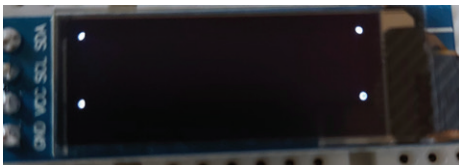
Listing 3: OLEDRect.py.

```
#-----
#
#      DISPLAY A RECTANGLE WITH TEXT INSIDE
#      =====
#
# In this progra a rectangle is displayed at the corners of the
# display and text R E C T A N G L E is displayed inside
#
# Author: Dogan Ibrahim
# File  : OLEDRect.py
# Date  : November.py
#-----
from PIL import Image, ImageDraw, ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)
font = ImageFont.load_default()

display.fill(0)
display.show()

width=(display.width)
height=(display.height)
image=Image.new('1', (width,height))
draw=ImageDraw.Draw(image)
draw.rectangle((0,0, 127, 31),outline = 255, fill = 0)
draw.text((15, 12),"R E C T A N G L E", font = font, fill = 255)
display.image(image)
display.show()
```



Figuur 6: We zien pixels in de vier hoeken van het OLED-display!



Figuur 7: De weergegeven tekst: 'ELEKTOR'. Natuurlijk!



Listing 4: OLEDShape1.py.

```
#-----
#
#   DISPLAY 4 RECTANGLES WITH NUMBERS 1,2,3,4
#   =====
#
# In this progra 4 rectangles are displayed. Number 1 is displayed
# inside rectangle 1, number 2 inside rectangle 2, and so on
#
# Author: Dogan Ibrahim
# File   : OLEDShape1.py
# Date   : November 2020
#-----
from PIL import Image, ImageDraw, ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)
font = ImageFont.load_default()

display.fill(0)
display.show()

width=(display.width)
height=(display.height)
image=Image.new('1', (width,height))
draw=ImageDraw.Draw(image)
draw.rectangle((0,0, 127, 31), outline = 255, fill = 0)
draw.line((64, 0, 64, 31), fill = 255)
draw.line((0, 16, 127, 16), fill = 255)
draw.text((32, 4), "1", font = font, fill = 255)
draw.text((94, 4), "2", font = font, fill = 255)
draw.text((32, 17), "3", font = font, fill = 255)
draw.text((94, 17), "4", font = font, fill = 255)
display.image(image)
display.show()
```



Figuur 8: Het beeld na het uitvoeren van OLEDrect.py.



Figuur 9: Het beeld na het uitvoeren van OLEDShape1.py (rechthoeken).



Figuur 10: De weergave na het uitvoeren van OLEDShape2.py (koorde en polygoon).



Listing 5: OLEDShape2.py.

```
#-----
#
#   DISPLAY A CHORD AND A POLYGON
#   =====
#
# In this program a chord and a polygon are drawn. The polygon is
# filled with white
#
# Author: Dogan Ibrahim
# File   : OLEDShape2.py
# Date   : November 2020
#-----
from PIL import Image, ImageDraw, ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)
font = ImageFont.load_default()

display.fill(0)
display.show()

width=(display.width)
height=(display.height)
image=Image.new('1', (width,height))
draw=ImageDraw.Draw(image)
draw.chord((0, 0, 90, 30), 10, 180, fill = 255)
draw.polygon([(100,5), (120,9), (120,25),
              (100,30)], outline=255, fill=255)
display.image(image)
display.show()
```

Vragen of opmerkingen?

Als u vragen of opmerkingen hebt over dit artikel, email dan de auteur via d.ibrahim@btinternet.com of Elektor via editor@elektor.com.

Met bijdragen van:

Tekst: Dogan Ibrahim
Redactie: Jan Buiting
Lay-out: Giel Dols
Vertaling: Evelien Snel

Het programma `OLEDRect.py` in **listing 3** demonstreert het tekenen van een rechthoek op de hoeken van de display en het schrijven van de tekst 'RECTANGLE' in die rechthoek. **Figuur 8** toont hoe dat er uitziet.

Het programma `OLEDShape1.py` in **listing 4** geeft vier rechthoeken weer. In rechthoek 1 wordt het getal '1' weergegeven. In rechthoek 2, wordt '2' weergegeven, enzovoort. **Figuur 9** toont hoe dat er uitziet.

Het programma `OLEDShape2.py` in **listing 5** geeft een koorde en een polygoon met vier hoeken weer. De parameter `fill` van de polygoon is ingesteld op 255, dus hij wordt gevuld met een witte achtergrond. **Figuur 10** toont hoe dat er uitziet.

Project 4: Een bitmap aanmaken en weergeven

We gaan nu een bitmap van 128×32 pixels aanmaken en die weergeven op het display. Het beeld dat we maken is een handgeschreven letter A. De stappen zijn als volgt:



Listing 6: OLEDBitmap.py.

```
#
#      DISPLAY BITMAP
#      =====
#
# This progra displays a bitmap image
#
# Author: Dogan Ibrahim
# File  : OLEDBitmap.py
# Date  : November 2020
#-----
from PIL import Image, ImageDraw, ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)

display.fill(0)
display.show()

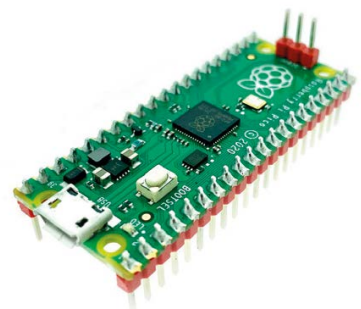
width=(display.width)
height=(display.height)
image=Image.new('1', (width,height))
image=Image.open('LetterA.png').resize((width,height),\
Image.ANTIALIAS).convert('1')
display.image(image)
display.show()
```

Advertentie

WORD ABONNEE EN ONTVANG

GRATIS Raspberry Pi + Headers

IEDERE 2 MAANDEN HET LAATSTE RASPBERRY
PI NIEUWS EN DE ALLERLEUKSTE PROJECTEN!



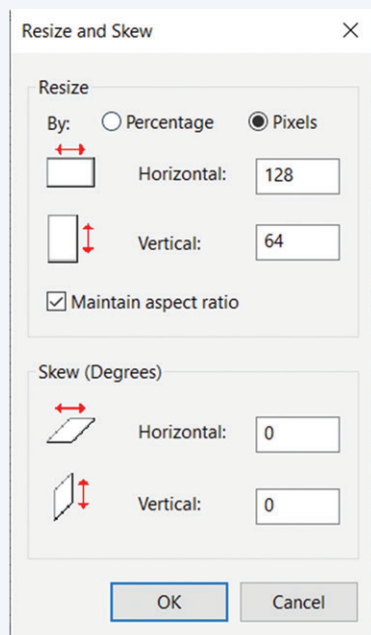
SLECHTS
€ 54,95
PER JAAR
(6 NUMMERS)

Uw voordelen:

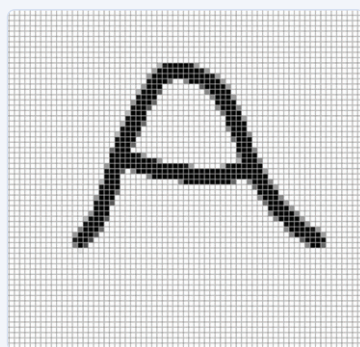
- Gratis Raspberry Pi + Headers
- Goedkoper dan zes losse nummers
- Iedere editie in de brievenbus; je hoeft de deur niet uit
- Elk nummer ook digitaal beschikbaar (PDF)
- Je leest MagPi al voordat het blad in de winkel ligt



ABONNEER NU OP WWW.MAGPI.NL



Figuur 11: Zet in MS Paint het aantal pixels op 128 x 64.



Figuur 12: Teken de gewenste vorm in MS Paint.



Figuur 13: Succes! Een min-of-meer handgeschreven letter op ons display.

Alle programma's in dit artikel zijn beschikbaar op de Elektor-webpagina bij het boek [1]. Scroll op die pagina omlaag naar *Downloads*, waar u een archiefbestand vindt met alles erin. Download de .zip file, pak hem uit op uw computer en zoek de zes hier besproken Python-programma's. ◀

210197-03

Aanmaken van het plaatje: We gebruiken het Microsoft-programma Paint om ons plaatje aan te maken.

Start MS Paint.

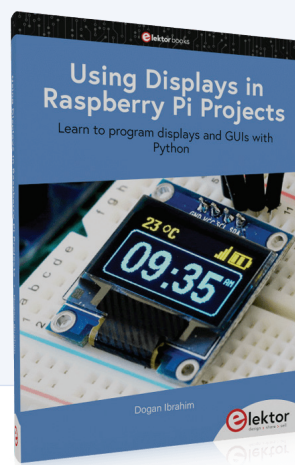
- Klik *Start Formaat wijzigen* en dan *Pixels*. Zet *Horizontaal* op 128 pixels en *Verticaal* op 64 pixels zoals in **figuur 11**. Klik *OK*.
- Klik op *Beeld* en dan op *Inzoomen* om de weergave te vergroten.
- Teken de vorm die u wilt weergeven met de muis. In dit project werken we met een handgetekende letter A zoals in **figuur 12**.
- Sla het bestand op als *LetterA.png*.
- Kopieer de file naar de homedirectory (/home/pi) op de Raspberry Pi. Dat kan bijvoorbeeld met het gratis programma WinSCP.

Het uiteindelijke programma, *OLEDBitmap.py*, is te zien in **listing 6**. De functie `Image.open` wordt gebruikt om de file te openen, dan wordt het formaat aangepast en hij wordt omgezet naar monochrome (1 bit per pixel) zoals de library nodig heeft. Het resultaat is te zien in **figuur 13**.



Gerelateerde producten

- **Boek: D. Ibrahim, *Using Displays in Raspberry Pi Projects* (Elektor 2021)**
www.elektor.nl/using-displays-in-raspberry-pi-projects
- **E-Book: D. Ibrahim, *Using Displays in Raspberry Pi Projects* (Elektor 2021)**
www.elektor.nl/using-displays-in-raspberry-pi-projects-e-book



WEBLINKS

- [1] **Download alle programma's:** www.elektor.nl/using-displays-in-raspberry-pi-projects
- [2] **PIL-Library:** <http://effbot.org/imagingbook/imagedraw.htm>